

一种并行混沌差异演化算法

郭振宇, 程博, 叶敏, 康龙云, 曹秉刚
(西安交通大学机械工程学院, 710049, 西安)

摘要: 在提出的并行混沌差异演化算法中, 差异演化算法与混沌搜索并行进行, 采用基于最优个体的差异演化算法, 差异演化的权重因子随种群的聚集度和算法停滞代数作动态调整, 而交叉率随种群的进化速度动态调整. 利用混沌的随机性和遍历性来扩大算法的搜索范围, 与差异演化算法实现功能互补, 在每一代并行计算后, 引入灾变算子, 在达到灾变条件时, 对种群实行灾变操作, 从而避免了算法陷入局部的最优值. 仿真结果表明, 与基本差异演化算法相比, 并行混沌差异演化算法的收敛速度快, 搜索效率高, 可用于解决复杂的机器学习问题.

关键词: 差异演化; 混沌; 灾变

中图分类号: TP18 **文献标识码:** A **文章编号:** 0253-987X(2007)03-0299-04

Parallel Chaos Differential Evolution Algorithm

Guo Zhenyu, Cheng Bo, Ye Min, Kang Longyun, Cao Binggang
(School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A novel parallel chaos differential evolution algorithm (PCDE) is presented, where differential evolution (DE) and chaos searching run concurrently, and in each iteration process the weighting factor is changed dynamically according to the current aggregation degree and number of stopping generations, the crossover factor is changed dynamically according to the current evolution rate. The randomness and space ergodicity of chaos mapping are considered to enlarge the searching range. The catastrophic factor is introduced to avoid premature convergence. DE and PCDE are tested with three well-known benchmark functions. The experiments show that PCDE is significantly superior to DE with higher efficiency to solve complicated optimization problems.

Keywords: differential evolution; chaos; catastrophe

差异演化算法^[1] (DE) 是一种基于种群中个体差异的演化算法, 该算法是由 Rainer 和 Kenneth^[2] 在 1996 年为求解切比雪夫多项式而提出的. 由于差异演化算法是基于种群中个体的差异进行进化的, 所以在种群差异度小时, 存在着搜索范围小和陷入局部极优值的问题. 混沌优化算法是一种新型的搜索优化算法, 由于混沌的随机性、遍历性特点, 使得混沌优化算法搜索范围较大. 本文对差异演化算法进行了改进, 与混沌优化并行结合, 实现了功能互补, 同时引入灾变算子, 增强算法跳出局部最优值的

能力, 从而提高了算法的搜索效率.

1 改进的差异演化算法

1.1 基本差异演化算法

差异演化算法是基于实数编码的进化算法, 最初的群体是随机产生的, 每个个体为搜索空间中的一个实向量. 令 $\mathbf{X}_i(g)$ 是第 g 代的第 i 个个体, $\mathbf{X}_i^L \leq \mathbf{X}_i(g) \leq \mathbf{X}_i^U$, 则

$$\mathbf{X}_i(g) = (x_{i1}(g), x_{i2}(g), \dots, x_{in}(g)) \quad (1) \\ i = 1, 2, \dots, M; g = 1, 2, \dots, t_{\max}$$

式中, $\mathbf{X}_i^U, \mathbf{X}_i^L$ 为个体的上、下界; M 为种群规模; $t_{\max}$ 为最大进化代数。

生成初始种群是在 n 维空间里随机产生 M 个个体, 实施措施为

$$x_{ij}(0) = x_{ij}^L + \text{rand}(0,1)(x_{ij}^U - x_{ij}^L) \quad (2)$$

式中: j 为第 i 个个体的第 j 个分量; $\text{rand}(0,1)$ 是 $[0, 1]$ 之间的随机小数。

变异操作是差异演化算法的关键步骤, 从种群中随机选择 2 个个体 x_{p_1} 和 x_{p_2} , 以及当前最优个体 $\mathbf{X}_b^{[3]}$, 且 $i \neq p_1 \neq p_2$, 则

$$h_{ij}(g+1) = x_{bj} + F(x_{p_1,j}(g) - x_{p_2,j}(g)) \quad (3)$$

式中: F 为权重因子; h_{ij} 为经差异操作产生的个体分量。

交叉操作可以增加种群的多样性, 其操作如下:

当 $\text{rand}(0,1) \leq C$, 或 $j = \text{rand}(i)$ 时

$$v_{ij}(g+1) = h_{ij}(g+1) \quad (4)$$

当 $\text{rand}(0,1) > C$, 或 $j \neq \text{rand}(i)$ 时

$$v_{ij}(g+1) = x_{ij} \quad (5)$$

式中: C 为交叉概率, $C \in [0, 1]$, 是 $\text{rand}(i)$ 在 $[1, n]$ 之间的随机数, 可以保证 $v_{ij}(g+1)$ 中至少有一个分量由 $h_{ij}(g+1)$ 贡献。

可以通过对评价函数进行比较选择向量 $\mathbf{v}_i(g+1)$ 和 $\mathbf{X}_i(g)$

$$\mathbf{X}_i(g+1) = \begin{cases} \mathbf{v}_i(g+1), & f(\mathbf{v}_i(g+1)) < f(\mathbf{X}_i(g)) \\ \mathbf{X}_i(g), & f(\mathbf{v}_i(g+1)) \geq f(\mathbf{X}_i(g)) \end{cases} \quad (6)$$

式中: f 为算法的适应度函数。演化算法程序反复执行式(3)~式(6), 直到达到最大进化代数, 或达到所要求的收敛精度。

1.2 改进策略

本文利用聚集度因子、进化停滞因子和进化速度因子对权重因子和交叉概率进行改进, 以提高算法的搜索效率。

1.2.1 聚集度因子 用聚集度因子来表示种群中个体的差异程度, 每代种群中个体的聚集度因子^[4]为

$$c = \min(f(g)_b, \overline{f(g)}) / \max(f(g)_b, \overline{f(g)}) \quad (7)$$

式中: $f(g)_b$ 表示第 g 代种群中最佳个体的适应度; $\overline{f(g)}$ 表示第 g 代种群的平均适应度。 $c \in [0, 1]$, 它反映出了种群当前的聚集程度, 同时也在一定程度上也反映了种群中个体的多样性。 c 值越大, 聚集程度越高, 个体差异性越小。

1.2.2 进化速度因子 进化速度因子为

$$a = \frac{f(g-2)_b - f(g-1)_b + C_{st}}{f(g-2)_b - f(g)_b + C_{st}} \quad (8)$$

式中: $f(g-1)_b$ 和 $f(g-2)_b$ 为第 $g-1$ 和第 $g-2$ 代的最优适应度; C_{st} 为很小的常数, 可使分子、分母恒不为 0。 $0 < a \leq 1$, 反映了种群的进化速度, a 值越小, 进化速度越快, 当经过一定代数之后, a 值保持为 1, 可判定算法停滞或者找到了最优解。

1.2.3 进化停滞因子 当连续 2 代甚至几代最优值都保持不变时, 说明算法处于停滞状态。停滞因子

$$s = \begin{cases} G_n \mu, & G_n \mu < s_{\max} \\ s_{\max}, & \text{其他} \end{cases} \quad (9)$$

式中: G_n 为停滞的代数; $s_{\max}$ 为停滞因子的最大取值; μ 为常数。

1.2.4 改进策略 差异演化算法的关键步骤是变异操作, 它由群体的差异向量信息来修正各个体的值, 当群体差异很小, 即群体聚集度很高时, 算法收敛速度急剧下降, 因此提出应根据种群的聚集度来调整权重因子, 使权重因子随种群聚集度的增大而增大, 随种群聚集度的减小而减小。当算法出现停滞时, 增大权重因子, 有利于种群的加速进化, 因此权重因子为

$$F = cF_s + s \quad (10)$$

式中: F_s 是变异率常数。

当 C 取值过大时, 可以加快收敛速度, 但是容易发生早熟现象; C 取值过小时, 则降低进化速度, 但不易使算法收敛于局部最优解。因此, 根据算法进化速度来动态地改变 C 的值, 当进化速度快时就减小 C 的值, 当进化速度变慢时, 甚至算法出现停滞时, 就增加 C 的值。 c 可表示为

$$C = aC_{Rs} \quad (11)$$

式中: C_{Rs} 是交叉率常数。

2 混沌搜索

混沌运动具有其独特的性质: ①随机性, 类似随机变量的杂乱表现; ②遍历性, 不重复历经一定范围内的所有状态; ③规律性, 由确定性的迭代式产生。基于最优个体的差异演化算法是各种模式差异演化算法中搜索速度最快的模式, 但是也更易于陷入局部极值, 而采用混沌搜索可以与其达到功能互补。

采用 Ulam-von-Neumann 混沌映射^[5]来实现混沌变异

$$p_{n+1} = 1 - 2p_n^2 \quad (12)$$

从种群中选择最优个体 $\mathbf{X}_i$, 将其映射到混沌遍历区

间,设 $x_{ij}(g)$ 为个体 i 的第 j 个分量,其取值范围为 $[a_j, b_j]$,根据式(13)、式(14)和式(15),将其映射到混沌遍历区间 $[-1, 1]$,得到

$$p_{ij}(g) = -1 + \frac{2}{b_j - a_j} x_{ij}(g) \quad (13)$$

$$p_{ij}(g+1) = f_{cs}[p_{ij}(g)] \quad (14)$$

$$x'_{ij}(g+1) = a_j + p_{ij}(g+1) \frac{b_j - a_j}{2} \quad (15)$$

式中: f_{cs} 代表式(12)的混沌映射公式。

将选择的个体各分量 x_{ij} ,按照式(12)进行 M 步的迭代,可得到一个规模为 M 的种群 A 。

3 灾变算子

在生物进化过程中,“灾变”就是外部环境的巨大变化,如冰河期、森林大火、大地震和瘟疫等,这是对绝大多数生物的灭顶之灾,并造成了大量个体的灭绝,只有少数适应能力特别强的个体得以生存。显然,灾变后幸存个体的生存能力更强,这对差异演化算法有很大启发。在获得某个最优解后,除了把最优解留下来以外,其他个体将重新随机产生,使得在较小的种群规模下获得较大规模的多样性,因此易于摆脱原先的局部最优解。

本文采用尖点灾变模型,其表达式为

$$V(x) = \frac{1}{4}x^4 + \frac{1}{2}ux^2 + vx \quad (16)$$

式中: x 是与当前群体进化有关的状态变量(差异演化机制和算子); u, v 为控制变量。

极值曲面满足

$$x^3 + ux + v = 0 \quad (17)$$

奇点满足

$$3x^2 + u = 0 \quad (18)$$

由式(17)和式(18),得到分歧曲线方程

$$4u^3 + 27v^2 = 0 \quad (19)$$

设 $f_m(t), f_a(t)$ 为 t 代种群的最优适应度和平均适应度,令

$$u \equiv [f_m(t-1) - f_m(t)]/f_m(t-1) \quad (20)$$

表示当前的相对进化程度。令

$$v \equiv [f_a(t) - f_a(t-s)]/s \quad (21)$$

表示近期 s 代平均进化速率($t > s$)。将式(20)和式(21)代入式(19),得到灾变的临界条件

$$f_a(t) - f_a(t-s) < -c \left(\frac{f_m(t)}{f_m(t-1)} - 1 \right)^{3/2} \quad (22)$$

$$c = 2 \times 3^{1/2} s/9$$

在并行算法搜索结束时,如果式(22)成立,即输出优化结果,反之则实行灾变,即保留当前最好的个体,随机产生其他个体,再次进行差异演化计算。

4 并行混沌差异演化算法结构

基于对原始差异演化算法及其改进策略、混沌搜索、灾变算子的讨论,本文提出了一种并行混沌差异演化算法(PCDE),如图1所示。

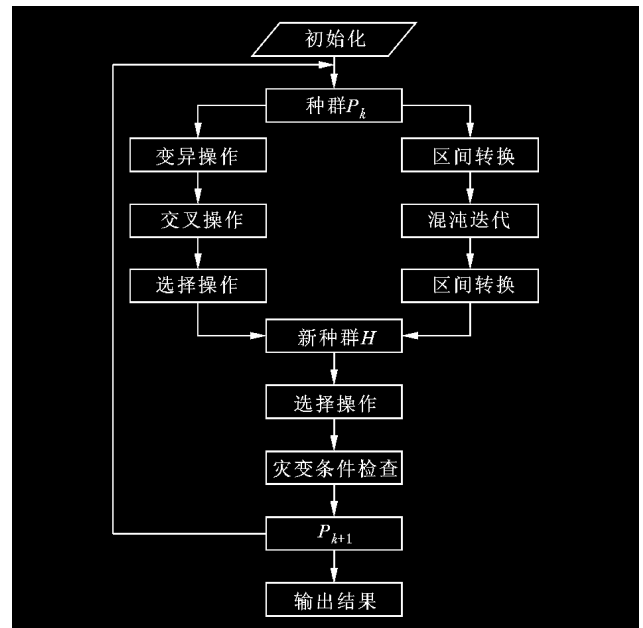


图1 算法流程图

将参数初始化和种群初始化后,得到种群 P_k ,并进行差异演化算法的变异操作、交叉操作和选择操作。同时,从 P_k 选择最优个体转化到混沌遍历区间,经 M 步混沌迭代后,得到种群 A ,再转化到优化变量值域后,与差异演化算法得到的种群合并,从而得到 H 。对 H 进行选择操作,保留适应度高的个体,并维持种群规模不变。对算法进行灾变条件检查,符合灾变条件时进行灾变操作,保留最优个体,其余个体随机产生做为 P_{k+1} ,不满足灾变条件则保留种群中的个体做为 P_{k+1} 。

5 模拟实验和结果分析

为了比较 DE 与本文提出的 PCDE 的性能,选择标准函数 Schaffer、Ackley 和 Schwefel 进行了实验测试。其中, Schaffer 和 Ackley 函数都是在原点取得最小值 0,而 Schwefel 函数则是在 $x_i (i=1 \sim n)$ 取 -420.9687 时取得最小值 0。

测试函数 1(Schaffer)

$$F(x) = 0.5 + \frac{\sin^2((x_1^2 + x_2^2)^{1/2}) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} \quad (23)$$

$$x_1, x_2 \in [-100, 100]$$

测试函数 2(Ackley)

$$F(x) = -20e^{-0.2 \left(\frac{1}{n} \sum_{i=1}^n x_i^2 \right)^{1/2}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)} + 20 + e \quad (24)$$

$$x_i \in [-10, 10], \quad n = 10$$

测试函数 3(Schwefel)

$$F(x) = 418.9829n + \sum_{i=1}^n x_i \sin(|x_i|^{1/2}) \quad (25)$$

$$x_i \in [-470, -370], \quad n = 10$$

选取最大迭代次数 N_u 为 2 000, DE 和 PCDE 各运行了 50 次, DE 和 PCDE 对 3 种函数的平均最优适应度 F_a 的比较如图 2~图 4 所示. 为了清楚地看出 DE 和 PCDE 在早期的收敛速度, 图中采用的横坐标为迭代次数的自然对数.

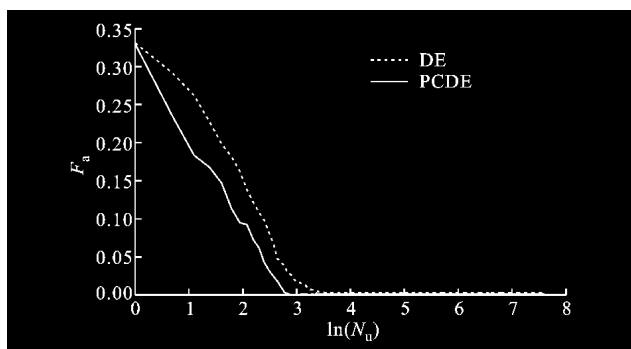


图2 Schaffer 函数平均最优适应度的进化曲线

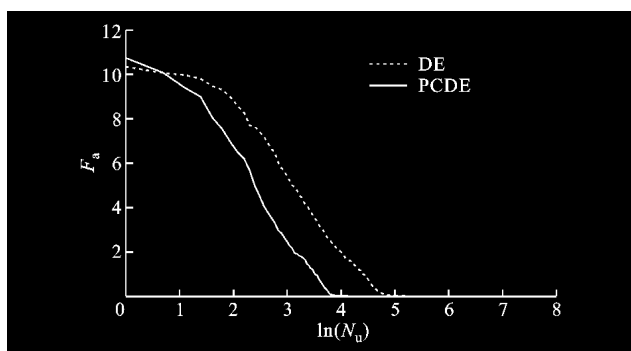


图3 Ackley 函数平均最优适应度的进化曲线

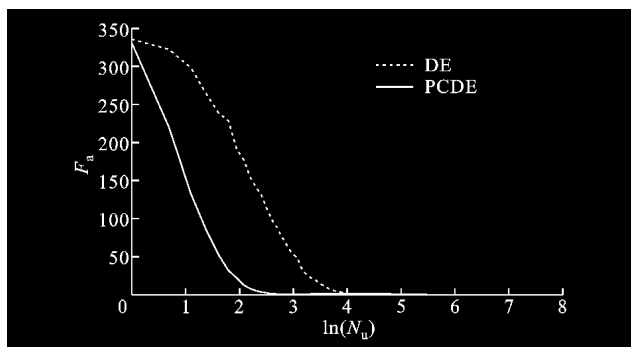


图4 Schwefel 函数平均最优适应度的进化曲线

从图 2~图 4 中可以看出, PCDE 在早期的收敛速度明显优于 DE, 适应度函数迅速下降, 而对于后期的收敛速度 2 种算法却相差不大.

DE 和 PCDE 在 2 000 步内没有收敛到指定精度的标记为失效(见表 1). 从表 1 中可以看出, PCDE 比 DE 的失效次数明显减少.

表 1 2 种算法的失效次数比较

算法	失效次数/次		
	Schaffer	Ackley	Schwefel
DE	2	6	4
PCDE	0	2	0

6 结 论

本文提出的差异演化算法是根据种群中个体的差异向量信息来完成进化的, 并根据种群的聚集度和停滞代数来动态地调整权重因子, 交叉概率则根据进化速度来动态地调整. 同时, 利用混沌的随机性和遍历性来进行混沌搜索, 对并行运算得到的结果进行了灾变条件的验证. 在算法停滞时, 对种群进行灾变操作, 通过对 Schaffer、Ackley 和 Schwefel 函数的计算表明, 与 DE 相比, PCDE 的失效次数明显减少, 收敛速度提高.

参考文献:

- [1] Storn R. Differential evolution design of an IIR-filter [C]// IEEE Int Conf on Evolutionary Computation. Los Angeles, USA: IEEE Press, 1996: 268-273.
- [2] Rainer S, Kenneth P. DE homepage [EB/OL]. (1997-02-11)[2006-08-15]. <http://www.icsi.berkeley.edu/~storn/code.html>.
- [3] Babu B V, Jehan M M L. Differential evolution for multi-objective optimization [J]. Evolutionary Computation, 2003, 4(12):2696-2703.
- [4] 张选平, 杜玉平, 秦国强, 等. 一种动态改变惯性权的自适应粒子群算法[J]. 西安交通大学学报, 2005, 39(10):1039-1042.
Zhang Xuanping, Du Yuping, Qin Guoqiang, et al. Adaptive particle swarm algorithm with dynamically changing inertia weight[J]. Journal of Xi'an Jiaotong University, 2005, 39(10):1039-1042.
- [5] 陈治飞. 混合混沌优化算法的研究及其在水下电机的应用[D]. 沈阳: 沈阳工业大学电气工程学院, 2003.

(编辑 管咏梅)